Automatic Paragraph Segmentation with Lexical and Prosodic Features

Automatic Paragraph Segmentation with Lexical and Prosodic Features

Catherine Lai[1]*, Mireia Farrús*[2]*, Johanna D. Moore*[1]

[1]School of Informatics, University of Edinburgh, Edinburgh, UK
[2]TALN Research Group, DTIC, Universitat Pompeu Fabra, Barcelona, Spain
clai@inf.ed.ac.uk, mireia.farrus@upf.edu, j.moore@ed.ac.uk

Abstract

As long-form spoken documents become more ubiquitous in everyday life, so does the need for automatic discourse segmentation in spoken language processing tasks. Although previous work has focused on broad topic segmentation, detection of finer-grained discourse units, such as paragraphs, is highly desirable for presenting and analyzing spoken content. To better understand how different aspects of speech cue these subtle discourse transitions, we investigate automatic paragraph segmentation of TED talks. We build lexical and prosodic paragraph segmenters using Support Vector Machines, AdaBoost, and Long Short Term Memory (LSTM) recurrent neural networks. In general, we find that induced cue words and supra-sentential prosodic features outperform features based on topical coherence, syntactic form and complexity. However, our best performance is achieved by combining a wide range of individually weak lexical and prosodic features, with the sequence modelling LSTM generally outperforming the other classifiers by a large margin. Moreover, we find that models that allow lower level interactions between different feature types produce better results than treating lexical and prosodic contributions as separate, independent information sources.

Index Terms: prosody, discourse, segmentation, paragraph, coherence

1. Introduction

Spoken presentations, such as video lectures, are becoming increasingly common sources of information. These audio documents often contain long spoken passages which are difficult to effectively browse and analyze without some notion of internal discourse structure. Previous studies on automatically detecting this sort of structure have tended to focus on broad topic or story level segmentation. However, fine-grained discourse segments, such as paragraphs, are also desirable for processing spoken language. From a browsing perpsective, paragraph segmentation is valuable for summarization [1] as well as improving readability of transcripts [2]. From a language understanding perspective, they provide a good test case for teasing out how subtler transitions in discourse structure are signalled using different aspects of speech. However, even though paragraph breaks are more readily available than other discourse structure annotations in the wild, little work has been done on automatic paragraph segmentation in text besides [1], let alone speech.

To address this gap, this paper investigates paragraph segmentation in a large corpus of TED talks.[1] In particular, we examine how lexical and prosodic features that help high level topic segmentation work at the paragraph level. Lexically based topic segmentation generally revolves around similarity-based notions of lexical coherence. In the much used TextTiling algorithm, for example, topic boundaries are determined by identifying points of low lexical similarity between consecutive windows in a text [3]. This basic approach has been improved by employing more abstract vector representations of the text [4, 5]. Other approaches, e.g. BayesSeg [6], further improve on this by modelling observed word distributions and segment boundaries via a Bayesian generative process on topics. While these word frequency based methods work for story level topic changes, performance worsens on spoken language [7], with a marked drop in performance for subtopic detection [8]. This suggests that we should look to other discourse oriented linguistic cues to obtain finer-grained segmentations.

Prime candidates for discourse structural markers are cue words and speech prosody [9]. Cue words such as 'because', 'well', and 'okay' have been associated with a broad range of functions related to discourse coherence [10, 11], and inclusion of cue word based features generally improves higher-level segmentation [6, 8, 12]. Similarly, prosodic features based on pitch, energy and timing have been used to perform topic segmentation on their own [13, 14, 15] or in conjunction with lexical features [8, 12, 16, 17]. While pause duration appears to be the most robust segmentation cue, paragraphs also seem to follow general prosodic declination and reset patterns [18]. So, we expect prosody to be informative of paragraph breaks.

While cue words and prosody are clearly helpful for segmentation tasks, how they can be best utilized is still an open question. Previous work incorporating lexical and prosodic features has suggested that they contribute independent evidence and, thus, can be modelled separately [16]. However, intonational analyses suggest that the prosodic form of lexical cues can also be important for interpretation [19]. In fact, combining individually weak lexical signals using AdaBoost improves text paragraph segmentation [1]. So, to take advantage of various subtle lexical and prosodic cues, we may need to combine evidence from different feature types at a relatively low level.

In the following, we investigate the predictiveness of cue words and supra-sentential prosody, compared with lexical coherence based features. As a baseline, we also examine the predictiveness of lexical features used by [1] for text segmentation. Our hypothesis is that discourse cues such as prosody and cue words are better indicators of paragraph breaks than traditional textual similarity or complexity measures. Beyond this we expect that modelling sequential information will improve performance, and that allowing low level interactions between lexical and prosodic features will produce better results than modelling these information sources separately. To test these hypotheses, we compare results from experiments using Support Vector Machines (SVMs), AdaBoost decision tree ensembles, and Bi-directional Long Short Term Memory (BLSTM) recurrent neural networks.

[1] http://www.ted.com

2. Experimental Setup

2.1. Data

In the following, we build paragraph boundary detectors based on a set of 1365 TED (Technology, Entertainment, Design) talks published before 2014. Talks are 15 minutes long on average and vary greatly in content and style. Most talks have one main speaker, although guests and audience members occasionally speak in some talks. The data set includes 1156 speakers of English with various accents, so that some of the speakers present more than one talk. Each talk is manually transcribed including punctuation and paragraph breaks. While there are no hard rules for determining paragraphs, transcribers attend to the audio stream when determining paragraph breaks.[2]

Altogether, the data set includes 151820 sentences and 20953 paragraphs, with an average of 7 sentences per paragraph. We obtain word timings through Viterbi forced alignment using an automatic speech recognition system. Sentences are detected based on punctuation using the Stanford CoreNLP sentence splitter [20]. We use the same toolkit to obtain Part-of-Speech (POS) tags, parse trees, and co-reference information. Word timings are then used to assign sentence boundary times. Given the aligned transcript, we extract various lexical and prosodic features as described in the following and summarized in Table 1.

2.2. Prosodic Features

F0 and intensity contours were extracted using Praat at 10 ms intervals with linear interpolation and octave jump removal for F0 [21]. For F0, parameter settings were automatically determined using the method described in [22]. F0 and intensity values were normalized over talks so that zero values represent speaker mean values: intensity measurements were normalized by subtracting the speaker mean for the talk, while F0 values were converted to log-scaled (semitone) values relative to speaker mean F0 value (Hz) to better fit pitch perception.

Based on previous analysis of paragraph prosody [18], we calculated aggregate statistics for each sentence: mean, standard deviation, maximum, minimum, median, slope, range (99th-1st quantiles). We also record the values for the previous and next sentences, as well as their differences to the target, and the difference between the first and last word of the target. For timing features (*dur*), we include the duration of the utterance, the number of words, the speaking rate (words/s), and the pause durations before and after the target sentence.

2.3. Lexical Baseline and Cue Words

We extract features based on those used for paragraph segmentation of texts in [1]. The features fall into three categories: surface form, syntactic form, and language model based complexity features. We also look at the predictiveness of POS tags [23]. Language models were estimated on training data using KenLM (1 to 5-grams) [24]. As in [1], these were used to estimate average word entropy and sentence probabilities. The individual features are listed in Table 1 (see [1] for details).

From these features we identify cue word related features. We record the first three words of the sentence (*w123*, 1-hot encoding). We also include binary indicators for the presence of any cue phrases at beginning, middle and end of the sentence from the list in [11] (*cwk*). We are specifically interested in the performance of these cue word encoding features relative to bag-of-words features over the entire sentence (*bow*).

[2] p.c. TED translation team.

	Features
dur	no. words, speaking rate, duration, prev/next pause durations
prosody	F0, intensity: mean, sd, max, min, slope, range for target, prev, next sentence, prev/next differences, dur
w123	1st, 2nd, 3rd word indicators (freq $>$ 100)
cwk	Knott [11] cue word at start, middle, end?
lm	average word entropy, sentence probability
syntax	no. phrases, parse tree top level children, branching factor, tree depth, cwk
pos	part-of-speech tag counts
bow	bag of words indicators
cw	w123, cwk
surface	no. words, relative position, final punctuation quote in previous, quote in target, quote incomplete, bow, w123
lex.base	pos, surface, lm, syntax
lex.coh	LDA, LSA, TF.IDF: target.sim, dscore, prev.sim, next.sim, lexical chain scores: lc.lemma, lc.sw, lc.gt1, lc.entity
lex.all	lex.coh, lex.base

Table 1: *Sentence level feature set guide.*

2.4. Lexical Coherence

To examine the performance of lexical coherence measures, we look at differences in topical and lexical similarity around potential boundary points based on Latent Dirichlet Allocation (LDA), Latent Semantic Analysis (LSA), and TF.IDF representations of the transcript. LDA and LSA models (100 topics) were trained on the whole dataset using gensim [25]. In the modelling stage, individual talks were treated as documents. The words in each document were lemmatized and words that occurred in more than half of the talks were excluded. Numeric vector representations were assigned to individual sentences using these models (100 dimensional vectors for LDA and LSA, 51503 dimensions, i.e., the vocabulary size, for TF.IDF).

As in TextTiling [3], we obtain similarity scores by summing sentence vectors falling inside the fixed windows before and after the target sentence, and then calculating the cosine similarity between these two vectors. We record the (moving-average) smoothed similarities (*target.sim*) as well as Texttiling depth scores (*dscore*). The latter measures the relative difference between the current similarity score and the closest 'peaks' in similarity to the left and right of the target sentence. A window size of 3 was used based on initial experiments using TextTiling for paragraph segmentation. We also include the cosine similarity of each sentence vector to the previous and next sentence for a more local measurement of lexical change (*prev.sim, next.sim*).

Besides topic model based features, we also measure coherence based on lexical chains. We calculate lexical chain cohesion scores by looking at the similarity at sentence breaks in terms of the lexical chains that span that boundary rather than lexical items in the surrounding windows. As in [12], chains are weighted by their inverse document frequency. We include separate features for chains based on all lemmas (*lc.lemma*), lemmas that occur more than once (*lc.gt1*), non-stopword lemmas (*lc.sw*), and chains based on automatically detected co-reference relations (*lc.entity*).

2.5. Evaluation Metrics

To evaluate our results we use standard discourse segmentation metrics: P_k [26] and WindowDiff (WD) [27]. Both metrics measure segmentation error using a sliding window (size k) through a document. For P_k, a penalty of 1 is added if a boundary is predicted for a no-boundary window or vice versa. For WD, a penalty of 1 is added if the predicted number boundaries does not match the ground truth. The summed penalties are normalized by the total number of windows to produce an error probability, with 0 indicating a perfect segmentation.

These standard metrics are known to be biased towards segmentations with very few predicted boundaries or edge clumping. Thus, following [28], we also report k-κ, a version of P_k which is explicitly corrected for chance agreement. We also extend the document sequence with k-1 zeros (no boundary) at either end to ameliorate the edge bias problem. We use k=3 following the standard practice of using half the average segment length for the dataset. For k-κ, scores of -1, 0, and 1 represent perfect disagreement, chance and perfect agreement respectively.

2.6. Classifiers

For reference, we provide results based on widely used unsupervised segmentation methods: BayesSeg [6] on raw text input, and TextTiling based on sentence bag of words. In both cases we allow the segmenter to automatically determine the number of boundaries. We also use TextTiling with prosodic sentence vectors and lexical chain scores. To help interpret P_k, WD and k-κ values, we give results for random and majority class segmentations [28]. To compare with the supervised approach of [1], we build classifiers using AdaBoost with Decision Tree estimators [29]. We also give results for linear SVMs [30] (cf. [18]). Both classifiers were built using Scikit-Learn [31].

To model sequential effects related to paragraph structure we also built LSTM recurrent neural network models [32], implemented using Keras [33] and Theano [34, 35]. Since features of interest potentially occur on both sides of a boundary, we use a single bi-directional LSTM layer to model the sequence both forward and backward in time, feeding into a softmax output layer (Figure 1a). Network parameters are optimized using AdaGrad with respect to cross-entropy loss. To prevent overfitting we include a dropout layer which randomly sets 30% of the hidden unit outputs to zero during training, and early stopping based on validation set loss. For training sequence input we use a centered window around a target sentence. We use predictions for all sentences in the window when calculating training and validation losses. However, we only consider the center target sentence output for our test set results. This was done based on early experiments that showed small windows provided better performance than using the whole talk sequence as input. This suggests that the relevant features for this sort of linear segmentation are relatively local. For brevity we only report results for window size 3.

We investigate lexical and prosodic feature fusion at different levels by modifying the BLSTM architecture as shown in Figure 1. The default mode is to concatenate all features as input to a single BLSTM (*feature fusion*). The other extreme is to train separate lexical and prosodic BLSTMs, adding an extra softmax layer to make the final decision on their separate class probability estimates (*decision fusion*). An intermediate version combines hidden outputs from separate lexical and prosodic BLSTM models before making the final decision (*intermediate fusion*, cf. score fusion in [36]).

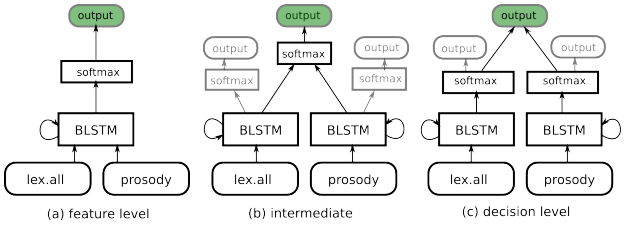

Figure 1: *BLSTM feature fusion at different levels.*

Classifier	P_k	WD	k-κ
Majority (no boundary)	0.41	**0.41**	0.00
Random	0.45	0.50	0.03
TextTiling+lc.gt1	0.44	0.45	0.01
TextTiling+bow	0.44	0.45	0.07
TextTiling+prosody	0.44	0.46	0.10
BayesSeg	**0.39**	0.47	**0.18**

Table 2: *Unsupervised baseline results.*

In the following, we discuss results from 10-fold cross validation with roughly 80/10/10 train, validation and test partitions. Model parameters for the different classifiers were tuned using validation set results from cross-validation.

3. Results

3.1. Unsupervised Baseline Experiments

Table 2 shows results for paragraph segmentations based on our unsupervised baselines. Interestingly, prosodic features provided the best performance out of our TextTiling variants for k-κ. Lexical chain based scores did not perform well with even the best chain source (lemmas occurring more than once) only performing a little above the random baseline. Overall, the Bayesian topic modelling approach (BayesSeg) performed the best in terms of k-κ and P_k. However, the poor WD score reflects a large number of false positives.

3.2. Supervised Classification Experiments

Table 3 shows the classification results for BLSTM models. In general, we see that the feature combinations perform better than their individual parts. For example, including features for both sentence initial words (*cw*) with the whole sentence bag of words (*bow*) produces better results than either set alone. Both of these feature sets give much better performance than the lexical coherence scores (*lex.coh*). This suggests that, unlike the coarser topic segmentation scenario, specific words or phrases are better indicators of paragraph boundaries than traditional lexical/topical similarity measures. In fact, when *cw* and *bow* features are removed from the *surface* set k-κ drops to 0.12.

Since the bag of words includes all words in the sentence, we also see that positional information is important beyond lexical identity. Word identity also seems more useful for this task than syntactic (*syntax*), part-of-speech (*pos*) and language model (*lm*) features. Even though these categories can encode some discourse related information, they are more useful in combination with other features. The same observation can be made for the prosodic features. That is, while pause duration is one of the stronger individual predictors of paragraph breaks, the combined prosodic feature set provides much better performance. Adding the low-performing coherence features to lexical features suggested in [1] makes an improvement (*lex.base* vs *lex.all)*. Overall, our best results come from the full combi-

Feature set	P_k	WD	k-κ
dur	0.38	0.39	0.13
prosody	*0.34*	*0.36*	*0.21*
lm	0.40	0.40	0.04
syntax	0.37	0.39	0.11
surface (-cw-bow)	0.37	0.39	0.12
pos	0.36	0.38	0.13
bow	0.36	0.38	0.17
cw	0.35	0.37	0.17
surface	*0.33*	*0.36*	*0.24*
lex.coh	0.38	0.39	0.10
lex.base	0.32	0.35	0.25
lex.all	*0.31*	*0.34*	*0.28*
cw+bow	0.34	0.37	0.21
cw+prosody	0.31	0.34	0.28
lex.all+prosody	**0.30**	**0.33**	**0.31**

Table 3: *BLSTM results for different feature sets: lower values are better for P_k and WD, higher values are better for K-κ.*

Feature set	SVM	AdaBoost	BLSTM
dur	0.10	0.13	0.13
prosody	*0.11*	*0.19*	*0.21*
lm	0.00	0.09	0.04
syntax	0.02	0.02	0.11
pos	0.02	0.02	0.13
bow	0.08	0.09	0.17
cw	0.09	0.07	0.17
surface	*0.11*	*0.14*	*0.24*
lex.coh	0.07	0.10	0.10
lex.base	0.13	0.16	0.25
lex.all	*0.14*	*0.17*	*0.28*
cw+bow	0.10	0.09	0.21
cw+prosody	0.13	0.21	0.28
lex.all+prosody	**0.17**	**0.26**	**0.31**

Table 4: *Supervised classification: k-κ results for SVM, AdaBoost, BLSTM models.*

nation of lexical and prosodic features (*lex.all+prosody*). That is, composing features at a low level gives us a stronger overall signal about discourse segmentation.

Table 4 shows k-κ results for BLSTM, AdaBoost, and SVM classifiers. Overall, AdaBoost performs better than the SVMs except with respect to *cw* based models. This once more suggests that combining various weak signals is necessary for this task. Again, we obtain the best performance from the full feature set for each classifier. However, it seems that the decision tree based AdaBoost gets more out of prosodic features than lexical features: the prosody-only AdaBoost classifier performs better than using all lexical features, and similarly, the duration model provides better results than the *cw+bow* model. Adding additional prosody and coherence features improves over *lex.base* (cf. [1]) by 10% absolute. The BLSTM classifiers generally perform better than AdaBoost over the different feature sets. These improvements show that sequence modelling is useful for this task, even when the sequences are short.

3.3. Incorporating Lexical and Prosodic Features

The results above are given for models which concatenate all features in a single input layer. Results for BLSTM models with fusion of lexical and prosodic features at different levels are shown in Table 5. We found that allowing feature level fusion performed better than simply combining class probabilities

Model	P_k	WD	K-κ
blstm:decision	0.31	0.34	0.27
blstm:feature	0.30	0.33	0.31
blstm:intermediate	**0.30**	**0.32**	**0.32**

Table 5: *Feature fusion at different levels.*

for separate lexical and prosodic models (decision level fusion). However, our best results come from the model where we train separate lexical and prosodic BLSTMs but concatenate the hidden layer outputs from these models to make the final decision (cf. Figure 1b). This shows that some amount of abstraction based on different feature sources may be helpful, but we don't want to treat lexical features and prosody as completely independent information sources.

4. Discussion and Conclusions

Beyond building a paragraph segmenter for speech, our goal is to better understand how linguistic devices are used to cue discourse structure in speech. The experiments described above confirmed our initial hypothesis that cue word and prosodic features are better indicators of paragraph structure than topical coherence measures. However, we can improve performance by allowing interactions between many individually weak lexical and prosodic signals. AdaBoost can do this to some extent, but BLSTMs go further by integrating sequence information, albeit with very small sequences in this case. The overall improved performance of the BLSTM can be attributed to being able to perform more informative composition across linguistic feature types and across time. Further work will look at expanding the contextual information available in the AdaBoost setup.

For the BLSTM models, the fact that we obtained better results using small windows rather than the full talk sequences suggests that short range dependencies are most important for this task. In fact, previous studies have suggested that the most useful prosodic features for segmentation are very close to the boundary [13]. However, segmentation results are still far from perfect, so further investigation into modelling long range dependencies is warranted. The current approach is likely limited by the amount of training data used. Future work will look at incorporating more data sources, as well as examining deeper BLSTM models and the contributions of forward and backward LSTM components. Framing the problem as a joint sentence and paragraph segmentation task may also help us incorporate useful sub-sentential prosodic and lexical knowledge more effectively. We also plan to look at topline human agreement on paragraph segmentation, and to extend this work to ASR output.

Although this paper only deals with linear segmentation, discourse structure is well understood to be hierarchical and multifaceted [37, 38]. While cue words have mostly been investigated as rhetorical connectives, they are also important for signalling hierarchical topic related structure [19]. Thus, we plan to investigate the relationship between cue words, prosody, and hierarchical structure in order to improve our segmentation, and to better shed light on the relationship between topic and rhetorically based notions of discourse structure.

5. Acknowledgements

We thank Peter Bell for providing the word timings. The second author is funded from the *EU's Horizon 2020 Research and Innovation Programme* under the GA H2020-RIA-645012 and the Spanish Ministry of Economy and Competitivity *Juan de la Cierva* program.

6. References

[1] C. Sporleder and M. Lapata, "Broad Coverage Paragraph Segmentation Across Languages and Domains," *ACM Trans. Speech Lang. Process.*, vol. 3, no. 2, pp. 1–35, 2006.

[2] A. Pappu and A. Stent, "Automatic Formatted Transcripts for Videos," in *Proc. Interspeech*, 2015.

[3] M. A. Hearst, "TextTiling: Segmenting Text into Multi-paragraph Subtopic Passages," *Computational Linguistics*, vol. 23, no. 1, pp. 33–64, Mar. 1997.

[4] M. Riedl and C. Biemann, "TopicTiling: A Text Segmentation Algorithm Based on LDA," in *Proc. ACL: Student Research Workshop*, 2012, pp. 37–42.

[5] L. Xie, L. Zheng, Z. Liu, and Y. Zhang, "Laplacian Eigenmaps for Automatic Story Segmentation of Broadcast News," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 1, pp. 276–289, Jan. 2012.

[6] J. Eisenstein and R. Barzilay, "Bayesian Unsupervised Topic Segmentation," in *Proc. EMNLP*, 2008, pp. 334–343.

[7] L. Du, W. L. Buntine, and M. Johnson, "Topic Segmentation with a Structured Topic Model." in *Proceedings of HLT-NAACL 2013*, 2013, pp. 190–200.

[8] P.-Y. Hsueh, J. D. Moore, and S. Renals, "Automatic Segmentation of Multiparty Dialogue." in *Proceedings of EACL 2006*, 2006.

[9] R. J. Passonneau and D. J. Litman, "Intention-based Segmentation: Human Reliability and Correlation with Linguistic Cues," in *Proc. ACL*, 1993, pp. 148–155.

[10] D. Schiffrin, *Discourse markers*. Cambridge University Press, 1987.

[11] A. Knott, "A Data-Driven Methodology for Motivating a Set of Coherence Relations," Ph.D. dissertation, University of Edinburgh, Edinburgh, Jul. 1996.

[12] M. Galley, K. McKeown, E. Fosler-Lussier, and H. Jing, "Discourse Segmentation of Multi-party Conversation," in *Proc. ACL*, Stroudsburg, PA, USA, 2003, pp. 562–569.

[13] E. Shriberg, A. Stolcke, D. Hakkani-Tür, and G. Tür, "Prosody-based automatic segmentation of speech into sentences and topics," *Speech Communication*, vol. 32, no. 12, pp. 127–154, 2000.

[14] G.-A. Levow, "Prosody-based Topic Segmentation for Mandarin Broadcast News," in *Proc. HLT-NAACL*, 2004, pp. 137–140.

[15] X. Wang, L. Xie, B. Ma, E. S. Chng, and H. Li, "Modeling broadcast news prosody using conditional random fields for story segmentation," *Proc. APSIPA*, pp. 253–256, 2010.

[16] G. Tür, D. Hakkani-Tür, A. Stolcke, and E. Shriberg, "Integrating Prosodic and Lexical Cues for Automatic Topic Segmentation," *Computational Linguistics*, vol. 27, no. 1, pp. 31–57, Mar. 2001.

[17] A. Rosenberg and J. Hirschberg, "Story Segmentation of Brodcast News in English, Mandarin and Arabic," in *Proc. HLT-NAACL*, 2006, pp. 125–128.

[18] M. Farrús, C. Lai, and J. D. Moore, "Paragraph-based Prosodic Cues for Speech Synthesis Applications," in *To Appear in the Proceedings of Speech Prosody 2016*, Boston, 2016.

[19] J. Hirschberg and D. Litman, "Empirical Studies on the Disambiguation of Cue Phrases," *Computational Linguistics*, vol. 19, no. 3, pp. 501–530, 1994.

[20] C. D. Manning, M. Surdeanu, J. Bauer, J. Finkel, S. J. Bethard, and D. McClosky, "The Stanford CoreNLP natural language processing toolkit," in *Proc. ACL: System Demonstrations*, 2014, pp. 55–60.

[21] P. Boersma, "Praat, a system for doing phonetics by computer," *Glot international*, vol. 5, no. 9/10, pp. 341–345, 2001.

[22] K. Evanini and C. Lai, "The importance of optimal parameter setting for pitch extraction." *The Journal of the Acoustical Society of America*, vol. 128, no. 4, pp. 2291–2291, Oct. 2010.

[23] J. Niekrasz, "Toward Summarization of Communicative Activities in Spoken Conversation," Ph.D. dissertation, University of Edinburgh, 2012.

[24] K. Heafield, "KenLM: faster and smaller language model queries," in *Proc. EMNLP Sixth Workshop on Statistical Machine Translation*, Edinburgh, Scotland, United Kingdom, July 2011, pp. 187–197.

[25] R. Řehůřek and P. Sojka, "Software Framework for Topic Modelling with Large Corpora," in *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*. Valletta, Malta: ELRA, 2010, pp. 45–50.

[26] D. Beeferman, A. Berger, and J. Lafferty, "Statistical Models for Text Segmentation," *Machine Learning*, vol. 34, no. 1-3, pp. 177–210, Feb. 1999.

[27] L. Pevzner and M. A. Hearst, "A Critique and Improvement of an Evaluation Metric for Text Segmentation," *Computational Linguistics*, vol. 28, no. 1, pp. 19–36, Mar. 2002.

[28] J. Niekrasz and J. Moore, "Unbiased discourse segmentation evaluation," in *Proc. SLT*, 2010, pp. 43–48.

[29] J. Zhu, H. Zou, S. Rosset, and T. Hastie, "Multi-class AdaBoost," *Statistics and its Interface*, vol. 2, no. 3, pp. 349–360, 2009.

[30] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin, "LIBLINEAR: A library for large linear classification," *Journal of Machine Learning Research*, vol. 9, pp. 1871–1874, 2008.

[31] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[32] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[33] F. Chollet, "Keras," https://github.com/fchollet/keras, 2015.

[34] F. Bastien, P. Lamblin, R. Pascanu, J. Bergstra, I. J. Goodfellow, A. Bergeron, N. Bouchard, and Y. Bengio, "Theano: new features and speed improvements," Deep Learning and Unsupervised Feature Learning NIPS 2012 Workshop, 2012.

[35] J. Bergstra, O. Breuleux, F. Bastien, P. Lamblin, R. Pascanu, G. Desjardins, J. Turian, D. Warde-Farley, and Y. Bengio, "Theano: a CPU and GPU math expression compiler," in *Proc. SciPy*, 2010.

[36] K. Nandakumar, Y. Chen, S. C. Dass, and A. K. Jain, "Likelihood ratio-based biometric score fusion," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 2, pp. 342–347, 2008.

[37] G. Redeker, "Ideational and pragmatic markers of discourse structure," *Journal of Pragmatics*, vol. 14, no. 3, pp. 367–381, 1990.

[38] M. Moser and J. D. Moore, "Toward a Synthesis of Two Accounts of Discourse Structure," *Computational Linguistics*, vol. 22, no. 3, pp. 409–419, 1996.